Management and Security Server Terminal ID Manager Guide

14.1.0

Table of contents

Terminal ID Manager Guide

Terminal ID Manager is a Management and Security Server Add-on product that provides IDs to client applications at runtime. This capability enables you to conserve terminal ID resources and significantly reduce operating expenses.

Specifically, you can:

- pool terminal IDs
- monitor ID usage
- manage inactivity timeout values for specific users

This guide describes the procedures to install and configure the Terminal ID Manager.

Setting up Terminal ID Manager

The Terminal ID Manager must be activated and configured before you can assign IDs to emulator sessions.

That is, at a high level, you must

1. Activate the Terminal ID Manager Add-on product.
2. Create a database.
3. Configure settings in the Terminal ID Manager Console.
4. Configure sessions to use Terminal ID Manager.
5. Assign those sessions to users or groups.

Follow the detailed steps in these sections:

System Requirements and Prerequisites

Before using Terminal ID Manager, verify that

- Management and Security Server is installed.
- The Terminal ID Manager Add-On activation file is available (if not already installed).
- Your emulator is supported.

Supported emulators and session types

Support for Terminal ID Manager is available in Host Access for the Cloud and Reflection for the Web for these web-based session types:

- IBM 3270, IBM 3270 Printer
- IBM 5250, IBM 5250 Printer
- ALC, Airlines Printer
- T27, T27 Printer
- UTS

Some Windows-based sessions also support Terminal ID Manager. Refer to your product's documentation.

Initial Setup

Initial Setup

Follow these steps to configure Terminal ID Manager for the first time:

- Install Terminal ID Manager (page 5)
- Open the Terminal ID Manager Console (page 5)
- Set up the Database (page 6)

Install Terminal ID Manager

Terminal ID Manager is an MSS Add-on product that is installed and enabled by default; however, it does require a separate activation file. The activation file can be installed as follows:

1. Log in to the MSS Administrative Console at `https://hostname/adminconsole`.
2. Click Configure Settings - Product Activation.
3. Click ACTIVATE NEW and browse to and click the activation file for the ID Manager:
   `activation.terminal_id_manager-<version>.jaw`

   The new product is added to the Product list.
4. Next step: Open the Terminal ID Manager Console (page 5) .

Open the Terminal ID Manager Console

Use the Terminal ID Manager Console to configure Server Settings and to Monitor ID pools.

1. Log in to the Terminal ID Manager Console at `https://hostname/tidm`.
2. Next step: Set Up the Database (page 6) .

Set Up the Database

1. Create an SQL file that will add your data definitions to the Terminal ID Management database:
   - Refer to the Example Scripts (page 8) .
   - The Populate IDs (page 9) example is a good place to start. Copy its contents into a local `.sql` file and open it in a text editor.
   - The script includes basic examples of the various ID attributes and pool types that you are likely to need.
   - After you identify your organizational needs, including the users and sessions that you plan to manage, edit your script as needed to define your pools.
2. In the Terminal ID Manager Console, use the Server Settings (page 7) panel to execute your script and configure the database:
   - In the Database management (page 8) section click the **Execute SQL File** button. Locate your script file and choose it for upload.
   - Click OK when prompted to confirm that you would like to configure the database.
   - A dialog will display the output of the script.
3. Upon successful execution, the Server Status (page 7) banner will immediately become green and indicate that the database is available.
4. Next step: Configure Additional Server Settings in the Terminal ID Manager Console (page 7) .

Server Settings

Server Settings

The Server Settings panel opens when you log on to the Terminal ID Manager Console.

Use this panel to configure your preferences for monitoring your terminal IDs.

- Server status (page 7)
- Server management (page 8)
- Database management (page 8)
- Server ID management (page 21)
- Logging and tracing (page 22)
- Change administrative password (page 23)

Server status

Note the Server database status in the top colored band, and what it means:

- **Server database available** (green)

  The Enable client requests box is checked, and Terminal ID Manager server is ready to be configured or used in sessions.

- **Server database not yet configured** (red)

  The server database has not yet been populated with data.

> **Tip**
>
> If you see this message, follow the steps to Set up the Database (page 6) .

- **Server database unavailable** (red)

  The server database is not functioning or cannot be accessed.

  To troubleshoot, review the `mss-mss-server` log files in the Cluster Management Console. To view the logs, open the Cluster Management console from the MSS Administrative Console drop-down menu. Open Services and click the `mss-mss-server` service. Then click ⋯ View Recent Logs or Download Logs.
  You may need to review how you Set up the Database (page 6) .

- **Client requests disabled** (orange)

  The database is configured, and the **Enable client requests** box is unchecked.

Server management

Enable client requests

Check this box if you want to process Terminal ID Manager requests from the client whenever the database is available.

Database management

Database management

The contents of the database can be updated using SQL commands.
Use the **Execute SQL File** button to upload a script file containing the commands to run.

See Example Scripts (page 8) for example template scripts to help get you started.

Example Scripts

Example Scripts

The Terminal ID Manager database can be maintained by uploading script files containing SQL commands which will be run against the database. The following scripts contain examples of common commands for initializing and maintaining the database.

Populate IDs (page 9) - Prepare the Terminal ID Manager for use

Check table utility (page 15) - Execute a check of all tables

Update tables (page 16) - Examples of various Terminal ID Manager database functions

Delete records (page 19) - Examples of commands to delete data

Populate IDs

```
-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |     ((>>---   SQL to prepare Terminal ID Manager for use. Tailor to your installation needs.   ---<<))       |
-- |                                                                                                               |
-- |        Lines beginning with "--" are comments; all others must be valid SQL for Terminal ID Manager.         |
-- |                                                                                                               |
-- |>=============================================================================================================<|

-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |  Do not modify or comment these statements - they obtain a connection to                                      |
-- |  the Terminal ID Manager Derby database, called DerbyIDM.                                                     |
-- |                                                                                                               |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|
SET SCHEMA IDM;

-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |                                ((>>---  Create Pool definitions   ---<<))                                     |
-- |  Column values:                                                                                               |
-- |                                                                                                               |
-- |  Poolname            -- Must be unique within the server.                                                     |
-- |  pooltype            -- Specify 1.                                                                            |
-- |  selectioncriteria   -- Encode the attributes that must be matched to get an ID from this pool.               |
-- |  heartbeatinterval   -- Number of seconds between the session sending an "I'm alive and using the ID"        |
-- |                         message to the server.                                                                |
-- |  commtimeoutinterval -- Number of seconds of non-communication before the server flags an ID with             |
-- |                         Timed-Out status.                                                                     |
-- |  hold                -- Set to 0. (Indicates 'in circulation'.  Admin hold in Monitor ID dialog box           |
-- |                         will set this value to 1.)                                                            |
-- |  update_ts           -- Initialize to today's date or a valid default, as indicated.  Indicates the last      |
-- |                         time this field was updated.                                                          |
-- |                                                                                                               |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|

insert into Pool
      (Poolname,        pooltype,           selectioncriteria,             heartbeatinterval, commtimeoutinterval, hold,
update_ts       )
VALUES
      ('Host3Sessions',    1,     '_SessType_SessName_HostAddr_HostPort',          60,              3600,            0,  '2010-07-06
10:00:00'),
      ('ClientIPAddrPool', 1,              '_SessType_ClientIP',                   70,              6000,            0,  '2010-07-06
10:00:00'),
      ('UserNamePool',     1,              '_SessType_UserName',                   90,              1200,            0,  '2009-12-10
10:00:00'),
      ('HostAddressPool',  1,              '_SessType_HostAddr',                   100,             360,             0,  '2009-12-10
10:00:00'),
      ('HostPortPool',     1,              '_SessType_HostPort',                   120,             200,             0,  '2009-12-10
10:00:00'),
      ('SessionPool',      1,              '_SessType_PoolName',                   180,             6000,            0,  '2009-12-10
10:00:00'),
      ('ClientDNSPool',    1,              '_SessType_ClientDNS',                  70,              6000,            0,  '2010-07-06
10:00:00'),
      ('4CriteriaPool',    1,     '_SessType_AssocSet_UserName_SessName',          180,             6000,            0,  '2009-12-10
10:00:00'),
      ('AllCriteriaPool',  1,     '_SessType_AssocSet_PoolName_UserName_SessName_HostAddr_HostPort_ApplName_ClientIP_ClientDNS',
                                                                                   180,             6000,            0,  '2009-12-10
10:00:00');


-- |>=============================================================================================================<|
-- | (Optional)  Display the Pools just added:                                                                     |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|

select * from Pool;




-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |                                ((>>---  Create AssociationSet definitions   ---<<))                           |
-- |                                                                                                               |
-- | This is an optional facility, and not useful to every installation of Terminal ID Manager.                    |
-- |                                                                                                               |
-- | An AssociationSet provides a mechanism to group together ID definitions that represent                        |
-- | host-defined resources or addresses.                                                                          |
-- |                                                                                                               |
-- |  The IDs in an AssociationSet typically:                                                                      |
-- |  ---------------------------------------                                                                      |
-- |  1. Are used at the same time.                                                                                |
-- |  2. Are used by the same end user.                                                                            |
-- |  3. Correspond to identifiers that are defined on one or more related hosts.                                  |
-- |  4. As a set, mirror some host-defined relationship between host resources; for instance, the host            |
-- |     relationship between a terminal and a set of printers.                                                    |
-- |                                                                                                               |
-- | To use this facility, first create an AssociationSet for a given association scenario that is to              |
-- | be used by a specific person. Give it a self-documenting name, such as "RJones-UTS-Gate27",  or               |
-- | "SF_Exec_Finance08".                                                                                          |
-- |                                                                                                               |
-- | Ensure the target IDs are created, and make them members of the AssociationSet by setting the name of that    |
-- | AssociationSet to the 'setname' field of each ID in the set.                                                  |
-- |                                                                                                               |
-- | Configure the end user emulation session's Connection Setup to use Terminal ID Manager, and in the            |
-- | ID Selection Setup dialog, select attributes as necessary to uniquely identify the target ID for each         |
-- | session, matching the choice of ID selection attributes to the 'selectioncriteria' of the pool that           |
```

```
-- | contains the target ID.                                                                                  |
-- |                                                                                                          |
-- | Example:                                                                                                 |
-- |                                                                                                          |
-- | Create an AssociationSet called "Payroll_Admin_4West" in order to provide a certain userX                |
-- | with a terminal ID associated with some printer IDs.                                                     |
-- |                                                                                                          |
-- | You might create a pool called "Set_Terminals" that has a Selection Criterial of Username, Session-name, |
-- | Session-type, and "ID association". You'd also create IDs in that pool with the correct attribute values |
-- | for each of their username, sessionname, and sessiontype fields. At least one of those IDs would contain |
-- | userX in its username field, and have a setname field value of "Payroll_Admin_4West".                    |
-- |                                                                                                          |
-- | Likewise, you could create a pool called "Set_Printers" with perhaps a Selection Criterial of Session-name, |
-- | Session-type, and "ID association".  You'd also create IDs in that pool with the correct attribute values |
-- | for each of their sessionname, and sessiontype fields, with the intended IDs for the user above having a |
-- | setname field containing 'Payroll_Admin_4West'.                                                          |
-- |                                                                                                          |
-- | Configure the Terminal ID Manager attributes for your terminal session to use Username, Session-name,    |
-- | Session-type, and ID association to match the selection criteria of the pools containing the terminal IDs. |
-- |                                                                                                          |
-- | Configure the Terminal ID Manager attributes for your printer sessions to use Session-name, Session-type, |
-- | and ID association to match the selection criteria of the pools containing the printer IDs.              |
-- |                                                                                                          |
-- | At runtime, userX would start a terminal session that would obtain the proper terminal ID, and it would  |
-- | appropriately be the one that is a member of the "Payroll_Admin_4West" AssociationSet.                   |
-- |                                                                                                          |
-- | When userX starts the printer session that has the _AssocSet attribute, Pool "Set_Printers" is           |
-- | searched (along with perhaps other pools) since it has the proper selection criteria. Since it           |
-- | contains IDs that match userX session attributes and belong to an AssoicationSet which are already       |
-- | allocated to 'userX', they would be the IDs that are provided during the ID request to the server.       |
-- |                                                                                                          |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|

insert into AssociationSet (    setname,         hold,       update_ts        )
                  VALUES  ( 'UTS_associate',     0,    '2009-12-10 10:00:00' );



-- |>=======================================================================================================<|
-- | (Optional)  display the Association-Sets just added:                                                     |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|

select * from AssociationSet;




-- |>=======================================================================================================<|
-- |                                                                                                          |
-- |                              ((>>---<>  Creating IDs -- The Basics   <>---<<))                           |
-- |                                                                                                          |
-- |  (A) All IDs are created in a Pool.                                                                      |
-- |                                                                                                          |
-- |  (B) The name of the ID must be unique within the Pool.                                                  |
-- |                                                                                                          |
-- |  (C) The combination of the idname and the poolname uniquely identifies an ID.                           |
-- |                                                                                                          |
-- |  (D) All IDs in a pool share these attribute values that are established on the pool definition:         |
-- |                                                                                                          |
-- |       1. selection criteria --                                                                           |
-- |          the attributes on the IDs that must match the attributes provided in the users                  |
-- |          request when the emulator session requests an ID from the Terminal ID Manager server.           |
-- |       2. the heartbeat interval defined on the pool.                                                     |
-- |       3. the communication timeout interval defined on the pool.                                         |
-- |       4. the value in their 'poolname' field.                                                            |
-- |                                                                                                          |
-- |  (E) IDs always require valid values for these fields, regardless of the selection criteria on the pool. |
-- |                                                                                                          |
-- |       1. idname      -- unique within the pool, as mentioned.                                            |
-- |       2. poolname    -- the same for all IDs in the pool. (Used by optional selection criteria 'PoolName') |
-- |       3. sessiontype -- the target host environment (in lowercase).                                      |
-- |                         (For required selection criteria '_SessType')                                    |
-- |       4. allocated   -- set to 0 (zero) to indicate the ID is currently not in use.                      |
-- |       5. timedout    -- set to 0 (zero) to indicate not in communication time-out with the server.       |
-- |       6. hold        -- set to 0 (zero) to indicate the ID is not out of operation due to               |
-- |                         Administrative 'hold'.                                                           |
-- |       7. update_ts   -- set to today's date, or some valid initialization value. (Field cannot be NULL). |
-- |                                                                                                          |
-- |  (F) Fields that must have valid values to match an optional 'selection criteria' defined on the pool    |
-- |                                                                                                          |
-- |       1. username      -- name of emulation user                (selection criteria includes '_UserName')  |
-- |       2. sessionname   -- name of the emulation session         (selection criteria includes '_SessName')  |
-- |       3. hostIP        -- the IP Address of the target host     (selection criteria includes '_HostAddr')  |
-- |       4. hostport      -- the port of the target host           (selection criteria includes '_HostPort')  |
-- |       5. application   -- host application to start on connect  (selection criteria includes '_ApplName')  |
-- |       6. clientaddress -- the IP Address of the emulation client (selection criteria includes '_ClientIP') |
-- |       7. setname       -- member of this AssociationSet         (selection criteria includes '_AssocSet')  |
-- |                           client request does not provide the name, just that it is to come from a set   |
-- |                                                                                                          |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|




-- |>=======================================================================================================<|
-- |                                                                                                          |
-- |                     ((>>---<>  Session Type Values (for sessiontype field)   <>---<<))                   |
```

```
-- |                                                                                                                |
-- |  Session Type in AWS                Value of sessiontype                                                       |
-- |  -------------------                ----------------                                                           |
-- |  Airlines Printer                   airlinesroute                                                              |
-- |  ALC                                alc                                                                        |
-- |  IBM 3270                           ibm3270                                                                    |
-- |  IBM 3270 Printer                   ibm3287                                                                    |
-- |  IBM 5250                           ibm5250                                                                    |
-- |  IBM 5250 Printer                   ibm3812                                                                    |
-- |  T27                                t27                                                                        |
-- |  T27 Printer                        t27printer                                                                 |
-- |  UTS Terminal                       uts                                                                        |
-- |  UTS INT1 Environment               uts                                                                        |
-- |                                                                                                                |
-- |----------------------------------------------------------------------------------------------------------------|



-- |================================================================================================================|
-- |                                                                                                                |
-- |                  ** NOTE **  There are limitations to the multi statement insert.                              |
-- |                                                                                                                |
-- |    The statements below insert multiple IDs, each bracketed by a pair of parentheses, separated by a comma.   |
-- |    A semi-colon terminates the statement and inserts multiple rows with a single commit operation to the DB.   |
-- |    However, because of a known bug in the Derby database engine (DERBY-1735), including too many inserts in   |
-- |    one statement creates a failure shown as "ERROR XJ001: Java exception: ': java.lang.StackOverflowError'."   |
-- |                                                                                                                |
-- |    Avoid the StackOverflowError by limiting the number of bytes in combined statements to about 275K bytes.   |
-- |    As a rough guide for limiting how many rows can be inserted in a single statement, two examples are:        |
-- |       -- 4000 insert rows where the rows average 70 bytes in length.                                           |
-- |       -- 1900 insert rows where the rows average 150 bytes in length.                                          |
-- |                                                                                                                |
-- |    The StackOverflowError will result in no rows being inserted.  You can generally fix and rerun the          |
-- |    failed statement by dividing it in half, terminating the first half with a semi-colon instead of the        |
-- |    comma, and inserting another copy of the first three lines of that statement before the second half.        |
-- |                                                                                                                |
-- |    For example:                                                                                                |
-- |        ...                                                                                                     |
-- |        ( the first half rows...   ),                                                                           |
-- |        (  'value-1',     'value-2',  'value-3', etc.... );       <= note ending semi-colon instead of comma.  |
-- |                                                                                                                |
-- |    insert into ID                                              <= repeat these 3 lines                         |
-- |          ( column-1,     column-2,   column-3,  etc.... )      <=         from the top                         |
-- |    values                                                      <=                 of the first half            |
-- |        (  'value-1',     'value-2',  'value-3', etc.... ),                                                     |
-- |        ( ...remaining 2nd half rows);                          <= 2nd half statement ending semi-colon        |
-- |                                                                                                                |
-- |----------------------------------------------------------------------------------------------------------------|



-- |================================================================================================================|
-- |                    Example ID definitions, showing various 'Selection Criteria' variations                     |
-- |================================================================================================================|
-- |                                                                                                                |
-- |  ((>>--- Example: IDs for Selection Criteria  '_SessType_ClientIP'  ---<<))                                    |
-- |                                                                                                                |
-- |----------------------------------------------------------------------------------------------------------------|

insert into ID
      ( idname,      poolname,         sessiontype, allocated, timedout, hold, clientaddress,  update_ts)
values
      ('idname771', 'ClientIPAddrPool', 'ibm3270',      0,        0,      0,   '15.23.5.8',    '2010-07-06 10:00:00'),
      ('idname772', 'ClientIPAddrPool', 'ibm3270',      0,        0,      0,   '110.42.2.14',  '2010-07-06 10:00:00'),
      ('idname773', 'ClientIPAddrPool', 'ibm3270',      0,        0,      0,   '21.2.25.9',    '2010-07-06 10:00:00');



-- |================================================================================================================|
-- |                                                                                                                |
-- |  ((>>--- Example: IDs for Selection Criteria  '_SessType_UserName'  ---<<))                                    |
-- |                                                                                                                |
-- |----------------------------------------------------------------------------------------------------------------|

insert into ID
      ( idname,         poolname,    sessiontype, allocated, timedout, hold,  username,      update_ts)
values
      ('idname121',  'UserNamePool',   'alc',         0,        0,      0,   'PeggySue',   '2010-07-06 10:00:00'),
      ('idname232',  'UserNamePool',   'alc',         0,        0,      0,   'BillyBob',   '2010-07-06 10:00:00'),
      ('idname343',  'UserNamePool',   'alc',         0,        0,      0,   'MaryJane',   '2010-07-06 10:00:00'),
      ('idname454',  'UserNamePool',   'alc',         0,        0,      0,   'SarahLee',   '2010-07-06 10:00:00'),
      ('idname565',  'UserNamePool',   'alc',         0,        0,      0,   'MollyAnn',   '2010-07-06 10:00:00'),
      ('idname676',  'UserNamePool',   'alc',         0,        0,      0,   'AnnaMarie',  '2010-07-06 10:00:00'),
      ('idname787',  'UserNamePool',   'alc',         0,        0,      0,   'JohnDoe',    '2010-07-06 10:00:00'),
      ('idname898',  'UserNamePool',   'alc',         0,        0,      0,   'BobbyJack',  '2010-07-06 10:00:00');



-- |================================================================================================================|
-- |                                                                                                                |
-- |  ((>>--- Example: IDs for Selection Criteria  '_SessType_HostAddr'  ---<<))                                    |
-- |                                                                                                                |
-- |----------------------------------------------------------------------------------------------------------------|

insert into ID
      ( idname,         poolname,       sessiontype, allocated, timedout, hold,   hostIP,               update_ts)
values
      ('idname987',  'HostAddressPool',   't27',         0,        0,      0,   'seamatt01.attm.com',  '2010-07-06 10:00:00'),
      ('idname654',  'HostAddressPool',   't27',         0,        0,      0,   '10.4.65.6',           '2010-07-06 10:00:00'),
```

```
      ('idname321',  'HostAddressPool',   't27',         0,        0,      0,   'hostfin1.jett.net',   '2010-07-06 10:00:00');



-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |  ((>>--- Example: IDs for Selection Criteria  '_SessType_HostPort'  ---<<))                                   |
-- |                                                                                                               |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|

insert into ID
      ( idname,         poolname,     sessiontype,    allocated, timedout, hold,  hostport,      update_ts)
values
      ('idname987',  'HostPortPool',   'ibm5250',         0,        0,      0,      24,      '2010-07-06 10:00:00'),
      ('idname654',  'HostPortPool',   'ibm5250',         0,        0,      0,      2424,    '2010-07-06 10:00:00'),
      ('idname321',  'HostPortPool',   'ibm5250',         0,        0,      0,      3800,    '2010-07-06 10:00:00');



-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |  ((>>--- Example: IDs for Selection Criteria  '_SessType_SessName_HostAddr_HostPort'  ---<<))                 |
-- |                                                                                                               |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|

insert into ID
      ( idname,      poolname,     sessiontype, allocated, timedout, hold, sessionname,  hostIP,     hostport,  update_ts)
values
      ('idname101', 'Host3Sessions', 'ibm3270',     0,        0,       0,  'HostXYZ01',  '10.2.25.4', 23,     '2010-07-06 10:00:00'),
      ('idname202', 'Host3Sessions', 'ibm3270',     0,        0,       0,  'HostABC01',  '10.2.17.1', 2323,   '2010-07-06 10:00:00'),
      ('idname303', 'Host3Sessions', 'ibm3270',     0,        0,       0,  'HostXYZ01',  '10.2.22.9', 29,     '2010-07-06 10:00:00');




-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |  ((>>--- Example: IDs for Selection Criteria '_SessType_PoolName'  ---<<))                                    |
-- |                                                                                                               |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|

insert into ID ( idname,            poolname,       sessiontype,       allocated,  timedout,    hold,         update_ts        )
        values
               ('idname001'      , 'SessionPool',    'ibm3270',            0,          0,         0,       '2010-07-06 10:00:00' ),
               ('idname002'      , 'SessionPool',    'ibm3270',            0,          0,         0,       '2010-07-06 10:00:00' ),
               ('idname003'      , 'SessionPool',    'ibm3270',            0,          0,         0,       '2010-07-06 10:00:00' ),
               ('idname004'      , 'SessionPool',    'airlinesroute',      0,          0,         0,       '2010-07-06 10:00:00' ),
               ('idname005'      , 'SessionPool',    'alc',                0,          0,         0,       '2010-07-06 10:00:00' ),
               ('idname006'      , 'SessionPool',    't27'    ,            0,          0,         0,       '2010-07-06 10:00:00' ),
               ('idname007'      , 'SessionPool',    'uts',                0,          0,         0,       '2010-07-06 10:00:00' ),
               ('idname008'      , 'SessionPool',    'ibm3812',            0,          0,         0,       '2010-07-06 10:00:00' ),
               ('idname009'      , 'SessionPool',    'ibm3270',            0,          0,         0,       '2010-07-06 10:00:00' );



-- |>=============================================================================================================<|
-- |                                                                                                               |
-- | ((>>--- Example: IDs for Selection Criteria '_SessType_ClientDNS' ---<<))                                     |
-- |                                                                                                               |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|

insert into ID ( idname,        poolname,      sessiontype, allocated, timedout, hold,       clientdns,             update_ts)
        values
               ('idname774', 'ClientDNSPool',   'ibm3270',     0,         0,      0,  'computerA.mydomain.com', '2010-07-06 10:00:00'),
               ('idname775', 'ClientDNSPool',   'ibm3270',     0,         0,      0,  'computerB.mydomain.com', '2010-07-06 10:00:00'),
               ('idname776', 'ClientDNSPool',   'ibm3270',     0,         0,      0,  'computerC.mydomain.com', '2010-07-06 10:00:00');



-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |  ((>>--- Example: IDs for Selection Criteria '_SessType_AssocSet_UserName_SessName'  ---<<))                  |
-- |                                                                                                               |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|

insert into ID
      ( idname,      poolname,   sessiontype, allocated,   timedout, hold, sessionname,    username,        update_ts,           setname)
values
      ('idname101', '4CriteriaPool', 'uts',        0,         0,       0,  'MySession1',  'PeggySue',    '2010-07-06 10:00:00',
'UTS_associate' ),
      ('idname202', '4CriteriaPool', 'uts',        0,         0,       0,  'MySession1',  'BillyBob',    '2010-07-06 10:00:00',
'UTS_associate' ),
      ('idname303', '4CriteriaPool', 'uts',        0,         0,       0,  'MySession2',  'MaryJane',    '2010-07-06 10:00:00',
'UTS_associate' ),
      ('idname405', '4CriteriaPool', 'uts',        0,         0,       0,  'MySession2',  'BobbyJoe',    '2010-07-06 10:00:00',
'UTS_associate' ),
      ('idname505', '4CriteriaPool', 'uts',        0,         0,       0,  'MySession3',  'SarahLee',    '2010-07-06 10:00:00',
'UTS_associate' ),
      ('idname606', '4CriteriaPool', 'uts',        0,         0,       0,  'MySession3',  'MollyAnn',    '2010-07-06 10:00:00',
'UTS_associate' );



-- |
>=============================================================================================================================
```

```
--
|
|
-- | ((>>--- Example: IDs for Selection Criteria
'_SessType_AssocSet_PoolName_UserName_SessName_HostAddr_HostPort_ApplName_ClientIP_ClientDNS' ---<<)) |
--
|
|
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - -|

insert into ID
      ( idname,      poolname,  sessiontype, allocated, timedout, hold,  sessionname, username, clientaddress,      clientdns,
hostIP,    hostport, application,    update_ts,    setname)
values
      ('idname101', 'AllCriteriaPool', 'uts',    0,       0,       0,  'MySession1', 'PeggySue', '10.2.2.1',  'computerA.mydomain.com',
'10.2.25.4',  12,   'Mapper', '2010-07-06 10:00:00', 'UTS_associate' ),
      ('idname202', 'AllCriteriaPool', 'uts',    0,       0,       0,  'MySession1', 'BillyBob', '10.2.2.2',  'computerB.mydomain.com',
'10.3.55.2',  12,   'Mapper', '2010-07-06 10:00:00', 'UTS_associate' ),
      ('idname303', 'AllCriteriaPool', 'uts',    0,       0,       0,  'MySession2', 'MaryJane', '10.2.2.3',  'computerC.mydomain.com',
'10.4.65.6',  12,   'Mapper', '2010-07-06 10:00:00', 'UTS_associate' ),
      ('idname404', 'AllCriteriaPool', 'uts',    0,       0,       0,  'MySession2', 'BobbyJoe', '10.2.2.4',  'computerD.mydomain.com',
'10.5.85.9',  12,   'Mapper', '2010-07-06 10:00:00', 'UTS_associate' ),
      ('idname505', 'AllCriteriaPool', 'uts',    0,       0,       0,  'MySession3', 'SarahLee', '10.2.2.5',  'computerE.mydomain.com',
'10.6.99.7',  12,   'Mapper', '2010-07-06 10:00:00', 'UTS_associate' ),
      ('idname606', 'AllCriteriaPool', 'uts',    0,       0,       0,  'MySession3', 'MollyAnn', '10.2.2.6',  'computerF.mydomain.com',
'10.7.11.1',  12,   'Mapper', '2010-07-06 10:00:00', 'UTS_associate' );





-- |>===========================================================================================================<|
-- | (Optional)  display the IDs, or the IDs in a particular Pool, or the IDs in a particular Association-Set    |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|

select * from ID;
select idname, poolname, sessiontype, sessionname,  hostIP, hostport from ID where POOLNAME = 'Host3Sessions';
select idname, poolname, sessiontype, username from ID where SETNAME  = 'UTS_associate';
select idname, poolname, sessiontype, setname, username, sessionname, hostIP, hostport, application, clientaddress, clientdns from ID where
POOLNAME = 'AllCriteriaPool';


-- |>===========================================================================================================<|
-- |  Always the final statement - do not Modify or comment.  This terminates the 'ij' command processor         |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|
EXIT;
```

Check Tables

```
-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |     ((>>---   SQL to execute a check of all tables in the Terminal ID Manager operational data   ---<<))     |
-- |                                                                                                               |
-- |        Lines beginning with "--" are comments; all others must be valid SQL for Terminal ID Manager.         |
-- |                                                                                                               |
-- |>=============================================================================================================<|

-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |  Do not modify or comment these statements - they obtain a connection to                                      |
-- |  the Terminal ID Manager Derby database, called DerbyIDM.                                                     |
-- |                                                                                                               |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|
SET SCHEMA IDM;

--  For a successful result, the utility will print out the following messages.
--  1
--  ----
--  1
--  1 row selected
--
--  If you obtain a different result, refer to the
--     Derby Server and Administration Guide, Version 10.5.


--  SELECT tablename, SYSCS_UTIL.SYSCS_CHECK_TABLE(
--  'IDM', tablename)
--  FROM sys.sysschemas s, sys.systables t
--  WHERE s.schemaname = 'IDM' AND s.schemaid = t.schemaid;
--  TABLENAME
--                              |2
--  ----------------------------------------------------------------------------------------------------
--
--  ASSOCIATIONSET
--                              |1
--  ID
--                              |1
--  IDM_VERSION
--                              |1
--  POOL
--                              |1
--
--  4 rows selected



SELECT tablename, SYSCS_UTIL.SYSCS_CHECK_TABLE(
'IDM', tablename)
FROM sys.sysschemas s, sys.systables t
WHERE s.schemaname = 'IDM' AND s.schemaid = t.schemaid;

-- |>=============================================================================================================<|
-- |  Always the final statement - do not Modify or comment.  This terminates the 'ij' command processor          |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|
EXIT;
```

Update Tables

```
-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |              ((>>--- Examples of various Terminal ID Manager database functions ---<<))                      |
-- |                                                                                                               |
-- |        Lines beginning with "--" are comments; all others must be valid SQL for Terminal ID Manager.         |
-- |                                                                                                               |
-- |>=============================================================================================================<|

-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |  Do not modify or comment these statements - they obtain a connection to                                      |
-- |  the Terminal ID Manager Derby database, called DerbyIDM.                                                     |
-- |                                                                                                               |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|
SET SCHEMA IDM;

-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |                   ((>>--- Example: Update one or more values of a pool definition  ---<<))                    |
-- |                                                                                                               |
-- |    Note:                                                                                                      |
-- |                                                                                                               |
-- |     * 'heartbeatinterval' and 'commtimeoutinterval' represent a number of seconds.  The Monitor IDs tab      |
-- |       shows those values in minutes (seconds divided by 60).  Since conversion of seconds to minutes         |
-- |       rounds down during division, 119 seconds will show as 1 minute, while 120 will show as 2 minutes.      |
-- |                                                                                                               |
-- |       If you set 'heartbeatinterval' to less than 60 seconds, a default minimum of                           |
-- |       60 seconds will be established for that value.                                                         |
-- |                                                                                                               |
-- |       If you set 'commtimeoutinterval' to less than 180 seconds, a default minimum of                        |
-- |       180 seconds will be established for that value.                                                        |
-- |                                                                                                               |
-- |       To allow 'heartbeatinterval' and 'commtimeoutinterval' for a pool to be controlled by the minutes      |
-- |       values specified on the Monitor IDs tab fields of "Heartbeat interval" or                              |
-- |       'Communication timeout interval', set the value on that pool field to -1.                              |
-- |                                                                                                               |
-- |     * When updating selectioncriteria on a pool definition, you must be certain that all IDs defined in      |
-- |       that pool have valid values specified for the attributes that are listed in the 'selectioncriteria'.   |
-- |                                                                                                               |
-- |       Example: if you change a selectioncriteria from '_SessType_SessName' to '_SessType_SessName_UserName',|
-- |       on the "PoolXYZ' pool, you must add valid values to the 'username' field of each ID in PoolXYZ.        |
-- |                                                                                                               |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|

--  display all the columns of the Pool table
SELECT poolname, pooltype, selectioncriteria, heartbeatinterval, commtimeoutinterval, hold, update_ts FROM Pool;


-- update all pools with the same values for heartbeatinterval and commtimeoutinterval

UPDATE Pool SET heartbeatinterval = 65, commtimeoutinterval = 960;



-- update pools 'HostPortPool' and 'ClientIPAddrPool' with the same heartbeat and comm. timeout values

UPDATE Pool SET heartbeatinterval = 91, commtimeoutinterval = 239 WHERE poolname = 'HostPortPool' or poolname = 'ClientIPAddrPool';



-- update pools 'UserNamePool' and '3270PoolTiny' to take the server default values for heartbeat and comm. timeout intervals

UPDATE Pool SET heartbeatinterval = -1, commtimeoutinterval = -1 WHERE poolname = 'UserNamePool' or poolname = '3270PoolTiny';




-- update pool 'UsersPool' to remove the _UserName attribute from its '_SessType_SessName_UserName' selection criteria
-- then remove the 'username' values from the IDs that are in pool 'UsersPool'

UPDATE Pool SET selectioncriteria = '_SessType_SessName' where selectioncriteria = '_SessType_SessName_UserName' and poolname = 'UsersPool';
UPDATE ID   SET username = null where poolname = 'UsersPool';



-- display the heartbeat and comm-timeout values for all pools

SELECT POOLNAME, HEARTBEATINTERVAL , COMMTIMEOUTINTERVAL  FROM POOL;




-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |                   ((>>--- Example: Update one or more values of an ID definition  ---<<))                     |
-- |                                                                                                               |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|


--  change the sessionname value on IDs for a particular user
update ID set SESSIONNAME = 'mySession' where USERNAME = 'userxyz' and POOLNAME = 'somepool';


-- add some IDs to an AssociationSet

update ID set SETNAME = 'myUTS' where IDNAME = 'myUTSID1';
update ID set SETNAME = 'myUTS' where IDNAME = 'myUTSID2';
update ID set SETNAME = 'myUTS' where IDNAME = 'myUTSID3';
```

```
-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |               ((>>--- Example: Update one or more values of an AssociationSet definition  ---<<))             |
-- |                                                                                                               |
-- |      Rename the name of an association set, and also rename the set reference to all the IDs in that set.     |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|


UPDATE ASSOCIATIONSET set SETNAME = 'UtsSet' where SETNAME = 'myUTS';
UPDATE ID set SETNAME = 'UtsSet' where SETNAME = 'myUTS';


-- |>=============================================================================================================<|
-- |  Always the final statement - do not Modify or comment.  This terminates the 'ij' command processor          |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|
EXIT;
```

Delete Records

```
-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |               ((>>---   SQL to delete Terminal ID Manager operational data   ---<<))                         |
-- |                                                                                                               |
-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |         This script contains sample statements to delete all or parts of the operational Pool,               |
-- |         AssociationSet, and ID data of the Terminal ID Manager's Derby database.                              |
-- |                                                                                                               |
-- |                                                                                                               |
-- |         This script assist users with deleting data, but does not remove the Pool, AssociationSet,           |
-- |         or ID tables from the database.                                                                       |
-- |                                                                                                               |
-- |                                                                                                               |
-- |         Note:                                                                                                 |
-- |                                                                                                               |
-- |          *  Names of IDs, Pools, and AssociationSets are case-sensitive.                                      |
-- |                                                                                                               |
-- |          *  Lines beginning with "--" are comments; all others must be valid SQL for Terminal ID Manager.     |
-- |                                                                                                               |
-- |          *  Uncomment only the SQL statements to be executed.                                                 |
-- |                                                                                                               |
-- |          *  A Pool can be deleted only if it does not contain any IDs.                                        |
-- |                                                                                                               |
-- |          *  To delete an AssociationSet, you must first change the setname of each ID that is a               |
-- |             member of that AssociationSet to NULL.                                                            |
-- |                                                                                                               |
-- |>=============================================================================================================<|

-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |  Do not modify or comment these statements - they obtain a connection to                                      |
-- |  the Terminal ID Manager Derby database, called DerbyIDM.                                                     |
-- |                                                                                                               |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|
SET SCHEMA IDM;


-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |              ((>>---  Statements to delete all ID, Pool, and AssociationSet data  ---<<))                    |
-- |                                                                                                               |
-- |        ==>  Execute these statements to empty the Terminal ID Manager database tables.                       |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|

-- removes all the ID and Pool and AssociationSet definitions

-- delete from ID;
-- delete from AssociationSet;
-- delete from Pool;




-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |       (>>---  Statements to remove or delete one or more IDs from Pools and AssociationSets  ---<<))         |
-- |                                                                                                               |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|

-- delete all the IDs contained in all of the Pools and AssociationSets

-- delete from ID;



-- delete all the IDs from one Pool.

-- delete from ID where poolname = 'Name_Of_Pool_to_Empty';



-- removes all the IDs currently members of an AssociationSet. Note that the IDs are not deleted.

-- update ID set setname = NULL where setname = 'Name_of_AssociationSet_to_empty';



-- moves all the IDs from one AssociationSet that are from a given Pool to a different AssociationSet.

-- update ID set setname = 'To_AssocSet' where setname = 'From_AssociationSet' and poolname = 'Name_Of_Pool';



-- delete all the IDs with a name matching a particular pattern where those IDs are from an AssociationSet
-- that has a name matching a particular pattern and are defined in a Pool whose name has a particular pattern.
-- For example:  this statement might delete IDs 'myTest1234' and 'newTest5678' if they are defined in Pool 'FinalTest'
-- and are members of the AssociationSet 'TestPayroll'.

-- delete from ID where idname like '%Test%' and setname like 'TEST%' and poolname like '%Test';
```

```
-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |                           ((>>---  Delete one or more Pool definitions   ---<<))                               |
-- |                                                                                                               |
-- |        ==>  Pool must not contain any IDs for delete to succeed                                               |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|

-- removes all the Pool definitions from Terminal ID Manager. Deleting all IDs empties all Pools so they can be deleted.

-- delete from ID;
-- delete from Pool;



-- remove one Pool definition from Terminal ID Manager.  Name of Pool is case-sensitive. Pool must be empty before deleting.

-- delete from ID   where poolname = 'Name_Of_Pool_To_Delete';
-- delete from Pool where poolname = 'Name_Of_Pool_To_Delete';




-- |>=============================================================================================================<|
-- |                                                                                                               |
-- |                       ((>>---  Delete one or more Association Set definitions   ---<<))                       |
-- |                                                                                                               |
-- |        ==>  Remove all IDs from an AssociationSet before the AssociationSet is deleted.  An ID is            |
-- |             removed from an AssociationSet by setting the 'setname' field in the ID to NULL.                  |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|

-- removes all the AssociationSet definitions from Terminal ID Manager

-- update ID set setname = NULL where setname IS NOT NULL;
-- delete from AssociationSet;



-- removes one AssociationSet definition from Terminal ID Manager.  Remove the IDs that are members of the AssociationSet first.

-- update ID set setname = NULL where setname = 'Name_Of_AssociationSet_To_Delete';
-- delete from AssociationSet   where setname = 'Name_Of_AssociationSet_To_Delete';



-- remove all AssociationSets with names matching a pattern.  Removes IDs from those AssociationSets first but
-- does not delete them.
-- example: will remove IDs from set membership, and then delete AssociationSets 'ABCzzzz123', 'zzzzXYZ', and '123zzzz'.

-- update ID set setname = NULL where setname like '%zzzz%';
-- delete from AssociationSet   where setname like '%zzzz%';



-- |>=============================================================================================================<|
-- |  Always the final statement - do not Modify or comment.  This terminates the 'ij' command processor          |
-- |- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -|
EXIT;
```

Server ID management

Review the server default values and change, as needed.

- Heartbeat interval

  Set the frequency (in minutes) that a client is expected to send a regular heartbeat message to the server. The message conveys that the ID is still in use by the client.

  This value applies to all pools unless the value is explicitly overridden for an individual pool in the Pools configuration.

- Communication timeout interval

  Set the number of minutes that must elapse without communication with an allocated session for an ID to be classified as timed out.

This value applies to all pools unless the value is explicitly overridden for an individual pool in the Pools configuration.

- Reclaim timed-out IDs automatically

  Specify the number of minutes before automatically reclaiming an ID that has been flagged as timed-out.

Logging and tracing

Select the log levels as needed. Keep in mind that performance may be affected when more information is logged.

Log	Description	Levels
Administrative activity	This refers to administrator actions, such as changing settings.	- Log errors and informational messages - Log errors - Disable logging
ID event activity	This refers to activity related to terminal IDs, such as clients requesting IDs.	- Log errors and informational messages - Log errors - Disable logging
General logging and tracing	This is used when analyzing server problems. Your Technical Support representative may request that this setting be changed to include debug information.	- Log errors, informational and debug messages - Log errors and informational messages - Log errors and warnings

Note

To view the logs, open the Cluster Management console from the MSS Administrative Console drop-down menu. Open Services and click the `mss-mss-server` service. Then click ▪▪▪ View Recent Logs or Download Logs.

Change administrative password

The Terminal ID Manager server administrator password permits access to the Terminal ID Manager Console, where the server settings are configured.

To change the current password (set during installation), enter the new password, and then enter it again to confirm it. Characters appear as dots.

Monitor IDs

Monitor IDs

The Monitor IDs panel displays a summary of terminal ID Pools and Association Sets.

First, note the server status. If the Server database is not available, review the `mss-mss-server` log files in the Cluster Management Console. To view the logs, open the Cluster Management console from the MSS Administrative Console drop-down menu. Open Services and click the `mss-mss-server` service. Then click ⋮ View Recent Logs or Download Logs.
You may need to review how you Set up the Database (page 6) .

Options

Display Options

Either keep the default to show All IDs, or use the drop-down menu to filter the display by the current ID attributes. Choose from these options:

- All IDs: the default
- Allocated IDs: only those pools that contain IDs that have been allocated.
- Non-allocated IDs: only those pools that contain IDs that have not been allocated
- Timed Out IDs: only those pools that contain IDs that have timed out.
- Held IDs: only those pools that contain IDs that are in Hold status.
- Timed Out and Held IDs: only those pools that contain IDs that are in a simultaneous status of being timed out and held.

Actions

> **Note**
>
> The Refresh button applies to both tables, while the Hold and Release buttons apply only to a selected pool.

- Refresh all settings on this panel.

  Click Refresh to redisplay the settings for each Pool and Association Set, including changes made on the individual panels.

- Hold all IDs in a particular pool.

  To set an administrative Hold on all of the IDs in a pool, click any cell in the row, except the underlined name of the pool. (Clicking the Name opens the details page for that pool.)

  Then click Hold, OK, and Close.

  Result: Any available IDs put on hold will not be considered for allocation. If an ID is currently allocated, the client will be notified that the ID should be returned.

- Release any IDs that are being Held in a particular pool.

  To Release IDs, click the Selection Criteria entry next to the Name of the pool. Then click Release, OK, and Close.

  Result: All of the IDs that were put on Hold are released and available for allocation, unless previously allocated.

To manage the individual IDs, click the Name of the Pool or Association Set, which open a separate panel.

When you select items to Hold or Release, you will see a message to confirm the action.

Pools

The currently defined pools are listed in this table, along with their attributes. Use the Column Chooser  to customize the display.

- Name: the name for this pool, set in the database
- Selection Criteria: the ID attributes that must match request attributes from the client in order to have an ID allocated from this pool
- ID Totals: the number of IDs in the pool
- Allocated: the number of IDs in the pool that are currently allocated

- Free: the number of IDs in the pool that are not allocated, and available for client sessions (if not held)
- Held: the number of IDs in the pool that are in Hold status, and therefore currently unavailable for further allocation to client sessions
- Timed Out: the number of IDs in the pool that have exceeded the configured communication timeout interval
- Communication timeout interval: the number of minutes that must elapse without communication with an allocated session for an ID to be classified as timed out

  If no value is specified here, the server default, specified in Server Settings is used. See Communication timeout interval.
- Heartbeat interval: the number of minutes in which a client is expected to send a regular heartbeat message to the server, indicating that the ID is still in use by the client.

The value set in Server Settings applies to all pools unless the value is explicitly overridden for an individual pool during pool configuration.

In the list of pools, you can

- Click Download CSV to view this table's data in a spreadsheet.
- Click the Name of a pool to configure settings for the IDs in that pool.

Pool IDs

The Pool IDs panel opens when you click the Name of a pool on the Terminal ID Manager - Monitor IDs panel. This panel shows the current Pool Settings, the ID Summary, and the specific IDs in that pool.

- Pool Settings

  Each ID in this pool satisfies the Selection Criteria, which was set in the database, and summarized here. The Communication timeout and Heartbeat intervals are shown here for easy reference.
- ID Summary

  You can easily see how many licenses in this pool are Allocated, Free, Held, or Timed Out. Look in the table to identify which IDs have those attributes.
- IDs

  By default, All IDs are shown in the table. Use the Show menu and the Column Chooser  to customize the display.

Note

An attribute without a value is either currently not defined or not required for the IDs in that pool.

ID	Description
Name	the name for this pool, set in the database
Session type	the type of session, such as 3270, UTS
Emulator type	Terminal or printer
Status	the status of the ID: Allocated, Free, Free but currently Held, Allocated but currently Held, Allocated and Timed Out, or Allocated and Timed Out and Held
Host port	the target host and port
Session name	the name of the client session that allocated the ID
User name	the user that is associated with the allocated session. The value for the user name is based on the user that is logged on to the operating system.
Client DN	the domain name of the client using the ID
Client IP	the IP address of the client using the ID
Association Set	If specified, this ID is a member of the Association Sets having this name
Allocated	a timestamp indicating when the ID was allocated
Heartbeat	a timestamp indicating the last communication from the client session
Timed out	a timestamp indicating when the ID went to a Timed Out status

ID	Description
Application	For UTS sessions, the application that the session will start after connecting

Actions

Use the control buttons to Hold, Release, or Reclaim one or more specific IDs.

- Hold: Set an administrative Hold on the selected IDs. The ID Status changes to Free/Hold or Allocated/Hold.

  Any available (Free) IDs that are put on Hold will not be considered for allocation. If an ID is currently allocated, the client will be notified (during the next client heartbeat) that the ID should be returned.
- Release: Release any of the selected IDs that are in Hold status. The ID Status changes to Free.
- Reclaim: This button displays if a pool is expanded to show IDs. When reclaimed, any selected ID that is allocated is changed to unallocated status.

Association Sets

An Association Set is a group of defined pools. The set provides a secondary grouping of IDs as another way to monitor IDs.

Each set of IDs mirrors an existing association between the device identifiers defined on the target hosts, in which groups of IDs operate together to provide a single emulation user a host-defined application functionality. For example, an association can facilitate routing of print output from a given emulation terminal to a specific emulation printer.

In the list of Association Sets, on the Terminal ID Manager - Monitor IDs panel, you can

- Click Download CSV to view this table's data in a spreadsheet.
- Click Show Set IDs (or the Name of a set) to configure settings for the Pool IDs in that set.

Set IDs

The Set IDs panel opens when you click the Name of an Association Set on the Terminal ID Manager - Monitor IDs panel. The Set IDs panel shows the ID Summary and the specific pool IDs in that set.

ID summary

At a glance, you can see how many licenses in this set are Allocated, Free, Held, or Timed Out. The Selection Criteria (defined in each pool) are not included.

Look in the table to identify which IDs have those attributes.

IDs

Each Pool in the set is listed with each of its IDs. By default, All IDs in the set are shown in the table. Use the Show menu and the Column Chooser  to customize the display.

- Pool: the name of a pool in this set
- ID: name of an ID in a named Pool
- User Name: the user that is associated with the allocated session. The value for the user name is based on the user that is logged on to the operating system.
- Emulator Type: Terminal or Printer
- Status: the status of the ID: Allocated, Free, Free but currently Held, Allocated but currently Held, Allocated and Timed Out, or Allocated and Timed Out and Held

Actions

Use the control buttons to Hold, Release, or Reclaim one or more specific IDs.

- Hold: Set an administrative Hold on the selected IDs. The ID Status changes to Free/Hold.
- Any available (Free) IDs that are put on Hold will not be considered for allocation. If an ID is currently allocated, the client will be notified that the ID should be returned.
- Release: Release any of the selected IDs that are in Hold status. The ID Status changes to Free.
- Reclaim: This button displays if a pool is expanded to show IDs. When reclaimed, any selected ID that is allocated is changed to unallocated status.

> **⚠ Caution**
>
> Use the Reclaim option with caution. Reclaiming an ID that is still in use can lead to synchronization problems.

When you select items to Hold, Release, or Reclaim, you will see a message to confirm the action.

On the Terminal ID Manager - Monitor IDs panel, click Refresh to update the table with your changes.

Configure and Assign Sessions

Configure and Assign Sessions

After Terminal ID Manager is installed and configured, you can configure sessions to use Terminal ID Manager. Then, assign the sessions to users or groups.

Configure the session

1. In the MSS Administrative Console, open Manage Sessions.
2. Either Add a session or Edit an existing one. Refer to the Supported emulators and session types, if needed.
3. In the session, locate the Use Terminal ID Manager setting, which is specific to your emulator.

   For instance, in a Host Access for the Cloud session:

Click Connection.

In the Device name menu, select Use Terminal ID Manager.

Configure the settings, and click Test. Refer to Help, as needed.

When done, click Save.

For other emulators, refer to your product documentation.

Assign the session

Remember to assign these sessions to specified users or groups.

To do so:

1. Return to the MSS Administrative Console.
2. Open Assign Sessions. Click Help, as needed.

Legal Notice

The only warranties for products and services of Open Text and its affiliates and licensors (“Open Text”) are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.